Jenkins 2.0

Jenkins 2.0 has been released! Please give it a try and provide your feedback.

Why

1. We need to claim our rightful place in Continuous Delivery. We have lots of pieces that address these modern needs, but we are not communicating this very well.
2. We need to revisit out of the box experience, so that Jenkins itself speaks that story and makes it clear what we are aiming for. Our software needs to speak for itself and tell people where we are going, so that the community can better focus efforts on the important parts.
3. And we need to do this while keeping what makes Jenkins great in the first place, which are the ecosystem, the community, and the extensibility that recognizes that one size does not fit all and let people do what they want to do.

Call to action: testing!

Rough Timeline

Here is a rough timeline of the release/launch:

- ~~February 29: Create "alpha" release~~ released
- ~~March 16: "feature complete" and cut alpha 4 release. Focus on making sure features landed work well~~ released
- ~~March 23: beta release with announcement, calls for testing to broader audience. Launch jenkins.io~~ released
- ~~April 1: Unscheduled Beta 2 with a number of bug fixes~~
- ~~April 6: Create first release candidate (RC)~~ released
- ~~April 20: Create 2.0 proper release with last RC~~
- ~~April 26: More coordinated launch marketing activities commence~~

Discussions

We'd like to validate that what we are proposing here actually solves people's problems. That will help us modify and prioritize the plan.

To collect those feedbacks in a manageable way, we have prepared a number of "2.0 planning tickets", divided under several themes called "epic tickets" in JIRA. This page and its children walk you through the main parts of those epics and details, and takes you to the actual tickets.

For historical archiving, here are earlier conversations that were done as mega threads:

- Mega thread
- Website thread and its Wiki page
- Reddit thread

As a community member and a future user, **we'd like YOU to COMMENT and VOTE on those tickets**. If you have more serious thoughts that do not fit in a comment or a vote, please feel free to post emails to the dev list or file additional tickets and link to the 2.0 planning tickets and/or epic tickets. The only thing we are trying to avoid is having more of the mega threads that nobody can keep up with.

Changes planned for 2.0

Overall pitch and its slide version.

User-visible changes

2.0 Pipeline as Code

Introduce a new subsystem in Jenkins that moves job configuration in SCM and makes job creation automatic.

Better out-of-the-box experience

A revamped new-user experience would help the numerous people who set up Jenkins for the first time. An important part is guiding new users through the installation of important plugins.

- Plugin Selection for the Setup Dialog
- JENKINS-31157

2.0 UX improvements

A more powerful plugin manager. A revamped UI for configuring jobs, views, build agents, etc. An easier to use "New Item/View/Agent" dialog.

- JENKINS-31095 — More consistent terminology
- JENKINS-32357 — Tabs on the job configuration page

2.0 Website

A new web site, which could include curated documentation, more visible events and blog, and/or a usable plugin index.

Internal changes

Servlet 3.1

Servlet 3.1 enables alternatives to polling Jenkins for changes every ten or so seconds through use of Web Sockets. This will require use of a fairly recent servlet container (Jetty 9.1+, Tomcat 8).

- JENKINS-23378 Raise minimum requirement to servlet 3.1

2.0 JavaScript Modularization

Emphasize UIs created in JavaScript, provide a simple framework for JS libraries that plugins can make use of.

Changes planned for a later release

Based on this discussion, the following proposals are currently considered for a release later than 2.0.

Configuration API

It's cumbersome to set up Jenkins in config management tools such as Chef. A proper API for configuring Jenkins would make this easier. The CLI isn't enough.

Proposal on the dev list

Storage Backend Changes

Being considered for a later 2.x release

Move parts of file storage into a database and/or make it pluggable.

Backward compatibility policy

Deprecated methods remain around forever, polluting code and documentation and making new development more cumbersome.

- JENKINS-31035

Move Groovy out of core

Enable more flexible upgrades of Groovy.

- JENKINS-29068 Split Groovy out of core

Hiding core libraries from plugins

Better isolation of the libraries core uses from plugins, so that "not breaking plugins" doesn't have to be a concern for use of libraries that should be internal to core, as it is today.

- JENKINS-30685 Hide core dependencies in plugin classpath